

(19)



(11)

EP 2 258 065 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
11.01.2017 Bulletin 2017/02

(51) Int Cl.:
H04L 1/00 ^(2006.01) **H03M 13/29** ^(2006.01)

(21) Application number: **09707145.0**

(86) International application number:
PCT/EP2009/000797

(22) Date of filing: **05.02.2009**

(87) International publication number:
WO 2009/098059 (13.08.2009 Gazette 2009/33)

(54) A METHOD AND SYSTEM FOR STOPPING EXECUTION OF A TURBO DECODER

VERFAHREN UND SYSTEM ZUM STOPPEN DER AUSFÜHRUNG EINES TURBO-DECODERS

PROCÉDÉ ET SYSTÈME POUR ARRÊTER L'EXÉCUTION D'UN TURBO-DÉCODEUR

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL
PT RO SE SI SK TR**

(30) Priority: **05.02.2008 EP 08290100**

(43) Date of publication of application:
08.12.2010 Bulletin 2010/49

(73) Proprietor: **Telefonaktiebolaget LM Ericsson
(publ)
164 83 Stockholm (SE)**

(72) Inventors:
• **CHARPENTIER, Sébastien**
F-14906 Caen (FR)
• **ANCORA, Andrea**
F-06300 Nice (FR)

(74) Representative: **Ericsson
Patent Development
Torshamnsgatan 21-23
164 80 Stockholm (SE)**

(56) References cited:
**EP-A2- 1 217 776 GB-A- 2 403 106
US-B1- 6 182 261**

EP 2 258 065 B1

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

FIELD OF THE INVENTION

5 **[0001]** The invention relates to the field of turbo decoders.

[0002] In particular, the invention relates to a method for stopping execution of a turbo decoder wherein said turbo decoder comprises a plurality of elementary decoders, wherein:

10 each respective one of the elementary decoders is configured for performing a respective sequence of decoding operations, and each respective one of the elementary decoders is arranged to receive a respective input from at least one other of the plurality of elementary decoders.

[0003] The invention also relates to a turbo decoder system employing a method for stopping the execution of a turbo decoder.

15 **[0004]** The invention also relates to software for configuring a data processing system.

BACKGROUND OF THE INVENTION

20 **[0005]** One class of error correcting codes is that of the so called turbo codes. A turbo code is decoded by a turbo decoder. Turbo decoders are characterized by comprising at least two elementary decoders who cooperate to produce decoded data.

[0006] Turbo codes are employed, e.g., in communications protocols, such as used for mobile telephony. For example, a sender, encodes original data using a turbo encoder for use during transmission, a receiver decodes received data using a turbo decoder. During transmission errors may occur. The use of a turbo encoder and turbo decoder allows the
25 correction of some errors. One protocol in which turbo codes are used is the High-Speed Downlink Packet Access (HSDPA) protocol.

[0007] Each elementary decoder takes as input the received data and the results of a previous elementary decoding operation, done by another elementary decoder. During execution the elementary decoders work on so-called soft data. Soft data gives probabilistic information on the data that was originally sent. In addition the soft data may comprise bit
30 reliabilities. The elementary decoders need to be able to use soft data, for example, SISO decoders (Soft Input Soft Output). The soft data can be transformed into hard data. Hard data consists of a specific guess of what the data, that was originally sent by the sender, was.

[0008] The structure and operation of turbo decoders is well known to a person skilled in the art. Turbo decoders are, for example, described in S. A. Barbulescu and S. S. Pietrobon, "Turbo codes: A tutorial on a new class of powerful
35 error correcting coding schemes, Part 1: Code structures and interleaver design", J. Elec. and Electron. Eng., Australia, vol. 19, pp. 129-142, Sep. 1999. and S. A. Barbulescu and S. S. Pietrobon, "Turbo codes: A tutorial on a new class of powerful error correcting coding schemes, Part 2: Decoder design and performance," J. Elec. and Electron. Eng., Australia, vol. 19, pp. 143-152, Sep. 1999.

[0009] A turbo decoder comprises an iterative algorithm comprising a number of iterations. Each iteration comprises the execution of each of the elementary decoders. The classical implementation of this iterative algorithm performs a
40 constant number of iterations independent of the channel condition, such as the signal to noise ratio. The ability to dynamically determine the number of iterations is called "Early stopping". The current state of the art comprises several techniques for early stopping.

[0010] One such approach for early stopping is described in US 7,093,180 B2. In US 7,093,180 B2 the turbo decoding is organized in a series of iterations. Each iteration consists of decodings by all the component decoders. At the end of
45 each full iteration it is determined if convergence or divergence has occurred. If so, the iteration is stopped.

[0011] A comprehensive overview of the state of the art of early stopping rules is provided by "Stopping Rules for Turbo Decoders"; TMO progress report 42-142; August 15, 2000; A. Matache, S. Dolinar and F. Pollara. Herein a turbo decoder with two elementary decoders is used. The report proposes a first method for early stopping. The first method
50 stops elementary decoding as soon as the outputs of the two elementary decoders are equal. This first method is applied after each elementary decoding. Also a second self-contained method for early stopping is proposed. The second method stops elementary decoding as soon as the second elementary decoder has converged; this rule is only applied after an elementary decoding done by the second elementary decoder.

[0012] EP1217776 (SAMSUNG ELECTRONICS Co Ltd) discloses an apparatus for stopping iterative decoding on a received frame comprised of information bits (associated with Log Likelihood Ratio) and then outputting the iteratively
55 decoding results.

[0013] It is a problem of the prior art that turbo encoders need many elementary decodings.

SUMMARY OF THE INVENTION

[0014] It is an object of the invention to reduce the number of elementary decodings that are needed by a turbo decoder.

[0015] This object is achieved by the method according to the invention, as defined in claim 1.

[0016] The inventors have recognized that a problem with the prior art lies in the granularity of applying stopping rules. When halfway in an iteration it could be determined that the execution of the turbo decoder can be terminated then it is possible to save one half of the iteration. This saving is especially non-negligible for HSDPA with a block size greater than 5000 bit. This type of savings is also especially relevant for devices with limited resources, such as mobile phones.

[0017] The advantages include reduced power consumption and reduced decoding latency. In this way a better compromise between latency and reliability of the decoded data is achieved.

[0018] The terminating is done if a number of sequences including the specific sequence has substantially converged or substantially diverged. This includes the method whereby only the specific sequences converges or diverges. This includes the method whereby the specific sequences converges or diverges and at least one other sequence converges or diverges. Instead of looking for converging or diverging the method can also only consider converging, or only consider diverging. The method could also look for one sequence that converges and one other sequence that diverges.

[0019] Considering the convergence or divergence of a single sequence has the advantage that less memory is needed.

[0020] Considering the convergence or divergence of at least two sequences has the advantage that the robustness of the method is increased, i.e., it becomes less likely that the termination is done in inappropriate situations.

[0021] The inventors have recognized that it is advantageous to stop the execution of a turbo decoder in case it is unlikely that the turbo decoder will eventually come up with a correct decoding. If two sequences corresponding to two elementary decoders have both converged, but not to the same data then the process is stable. In that case it is not likely that more progress towards a correct output will be made. By having the turbo decoder reporting this eventuality, a communication system comprising the turbo decoder can take further action. For example, the communication system could request the resending of the data, or it could terminate the application depending on the data. In case the turbo decoder uses more than 2 elementary decoders the reliability is increased when stopping is done in case of convergence of more than 2 sequences corresponding to elementary decoders.

[0022] The advantages of this method include a faster response to situations where the turbo decoder fails for the above reason, and reduced power consumption in that situation.

[0023] It is most efficient if the method of claim 3 is applied after each decoding operation. In that case it is detected at the earliest that a correct decoding has become unlikely, but it also possible to apply the method of claim 3 only at the end of an iteration.

[0024] It can be convenient to combine the step of determining if a first sequence of decoding operations has converged to a first data in claim 3 with the step of determining if the specific sequence of decoding operations of the specific elementary decoder has substantially converged or substantially diverged in claim 1.

[0025] A preferred embodiment of the method according to the invention is characterized by the method of claim 4.

[0026] The inventors have recognized that there is no need to restrict oneself to one type of rule; rather it brings advantages to combine different rules. Since testing for termination is relatively cheap compared to an extra elementary decoding operation, it is opportune to test more than one criterion to maximize the probability of early stopping.

[0027] Determining if a first sequence of decoding operations substantially agrees with a second sequence of decoding operations can first involve determining if the first sequence converges; it can also involve testing if the second sequence converges. When determining substantial agreement involves determining convergence the method of claim 4 can conveniently be combined with any one of the method of claims 1 and 3. By combining two determinations for convergence into one step the advantage of faster execution is achieved, also the amount of software or hardware to execute the method is reduced, also the complexity of such software or hardware is reduced.

[0028] In particular the combination of claim 3 and 4 is especially advantageous.

[0029] The turbo decoder system according to the invention is characterized by claim 8.

[0030] The turbo decoder system can be configured to use the method using software, or the turbo decoder system may comprise hardware, such as electronic circuits that are configured to perform the method, or the turbo decoder system can be controlled using software, or the turbo decoder system may comprise a combination of possibly dedicated hardware and software to implement the method of the invention. A single device could comprise such a turbo decoder system. On the other hand such a turbo decoder system could also comprise a plurality of separate devices.

[0031] The software according to the invention is characterized by claim 9.

[0032] The scope of claim 9 includes software comprising a number of processing means, such that each step of a method according to the invention can be achieved by at least one of the number of processing means. In one embodiment each specific processing means of the number of processing means corresponds to a specific step of the method according to the invention. The programming means can be fabricated using various well known high-level programming languages, such as, C, C++ or Pascal. The programming means can alternatively be fabricated using low-level programming languages such as assembly, machine codes or microcode. A processing means can correspond to an

architectural structure, such as a subroutine, or an object, but this is not necessary, and a processing means can also correspond to one or more lines of programming code, or part thereof. A processing means could also be comprised in a larger processing means addressing an ulterior goal of the software.

[0033] Embodiments that do not fall under the scope of the claims are examples useful for understanding the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0034] The invention is explained in further detail, by way of example and with reference to the accompanying drawings, wherein:

Fig. 1 is a block diagram showing an application of the turbo decoder system according to the invention into a communication system.

Fig.2 is a block diagram illustrating a first embodiment of the turbo decoder system according to the invention.

Fig. 3 is a block diagram illustrating a second embodiment of the decoder system according to the invention.

Fig.4 is a block diagram illustrating an embodiment of a stopping unit.

Fig. 5 is a flowchart illustrating a first embodiment of the method for stopping execution of a turbo decoder according to the invention.

Fig.6 is a flowchart illustrating a second embodiment of the method for stopping according to the invention.

Fig. 7 is a flowchart illustrating a third embodiment of the method for stopping according to the invention.

Fig. 8 gives various equations representing various aspects of embodiments of the method for stopping according to the invention.

Fig.9 gives further equations representing various aspects of embodiments of the method for stopping according to the invention.

[0035] Throughout the Figures, similar or corresponding features are indicated by same reference numerals.

DETAILED EMBODIMENTS

[0036] While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail one or more specific embodiments, with the understanding that the present disclosure is to be considered as exemplary of the principles of the invention and not intended to limit the invention to the specific embodiments shown and described.

[0037] For convenience, the method of stopping a turbo decoder is described for a turbo decoder having two elementary decoders; however, the method is applicable to any turbo decoder having at least two elementary decoders.

LEGEND FOR THE FIGURES

[0038]

Reference Y denotes Yes.

Reference N denotes No.

Reference Fig. denotes Figure.

Reference numeral (100): a communication system.

Reference numeral (102): a sender.

Reference numeral (104): a receiver.

Reference numeral (106): an original data.

Reference numeral (108): a turbo decoder.

Reference numeral (109): a transmitted data.

Reference numeral (110): a received data.

Reference numeral (112): a turbo decoder system.

Reference numeral (114): a decoded data.

Reference numeral (202): an elementary decoder 1.

Reference numeral (204): an elementary decoder 2.

Reference numeral (206): a stopping decision unit.

Reference numeral (300): a hard data unit 1.

Reference numeral (302): a hard data unit 2.

Reference numeral (400): a memory 1.

Reference numeral (402): a memory 2.

Reference numeral (404): a stopping decision logic.

Reference numeral (406): an output of elementary decoder 1.

Reference numeral (408): an output of elementary decoder 2.

Reference numeral (501): Perform a specific elementary decoding operation.

Reference numeral (502): Did the specific sequence converge or diverge?

Reference numeral (504): Terminate the execution of decoding operations.

Reference numeral (602): Did the specific sequence converge?

Reference numeral (604): Did the sequence prior to the specific sequence converge?

Reference numeral (606): Did both sequences converge to the same data?

Reference numeral (608): Report failure.

Reference numeral (702): Does the specific elementary decoding operation substantially agree with the previous one?

Reference numeral (704): Other reasons for stopping?

Reference I in Equation (902): After a decoding operation performed by elementary decoder 1 (202).

Reference II in Equation (902): After a decoding operation performed by elementary decoder 2 (204).

[0039] It is first described, and illustrated in **Fig. 1**, how a turbo decoder system (112) employing the method could be used in a communication system (100), by two entities, viz. a sender (102) and a receiver (104). The sender (102) selects an original data (106) and causes it to be used as input for a turbo encoder (108). The turbo encoder (108) computes a transmitted data (109) that is suitable to be used in a communication system, such as a wireless communication system, such as HSDPA, or in a wired communication system, such as a digital network, such as the Internet. The transmitted data (109) is transmitted to the receiver (104). The receiver (104) receives received data (110) and causes it to be used as input for a turbo decoder system (112). The turbo decoder system (112) computes a decoded data (114).

[0040] If no errors occur during the transmission of the transmitted data (109), the received data (110) will be equal to the transmitted data (109). However, if the transmission is noisy, errors will occur and the received data (110) will differ from the transmitted data (109).

[0041] The turbo decoder attempts to reconstruct the original data (106). If no errors or fewer errors than the turbo decoder system (112) can correct have occurred during transmission then the decoded data (114) will be equal to the original data (106).

[0042] It can happen that the turbo decoder fails, for example, if too many errors have occurred and the received data (110) differs too much from the transmitted data (109). In that case the decoded data (114) will be unequal to the original data (106). Even if few errors have occurred there is a chance that decoded data (114) is unequal to the original data (106).

[0043] The turbo encoder (108) comprises a plurality of elementary encoders. Each elementary encoder corresponds to a corresponding elementary decoder in the turbo decoder system (112). Each elementary encoder in the turbo encoder (108) adds error correcting data (e.g. parity data) aimed at the corresponding elementary decoder of the turbo decoder system (112). Typically the turbo encoder (108) will also add data aimed at all elementary decoders, typically the latter data will take the form of systematic data, which is representative of the original data, such as a copy, or permutation of the original data. For convenience of exposition it is assumed that the turbo encoder adds systematic data, although this does not restrict the method. The error correcting data and any other data is typically interleaved.

[0044] Typically a turbo encoder (108) contains two elementary encoders; typically these elementary encoders are of the convolutional encoder type. Typically a turbo decoder system (112) contains two elementary decoders; typically these elementary decoders are of the convolutional decoder type. Typically these elementary decoders are also of SISO type.

[0045] Turbo decoders can also be employed in a storage system. Instead of transmission of the transmitted data (109) the data is stored in a suitable digital storage medium. The receiver (110) can read the received data from the storage medium. In this case the receiver (104) and the sender (102) may be the same entity, although this is not necessary.

[0046] As a further example a turbo decoder system (112) can be used in conjunction with digital video broadcasting, satellite communications and space exploring systems.

[0047] In **Fig. 2** a block diagram is shown, illustrating an embodiment of a turbo decoder system (112). Two elementary decoders: elementary decoder 1 (202) and elementary decoder 2 (204) are arranged to receive a received data (110). Elementary decoder 2 (204) is arranged to receive input from elementary decoder 1 (202) and elementary decoder 1 (202) is arranged to receive input from elementary decoder 2 (204). A stopping decision unit (206) is arranged to receive input from elementary decoder 1 (202) and elementary decoder 2 (204).

[0048] The stopping decision unit (206) is capable of terminating elementary decoder 1 (202) and elementary decoder 2 (204).

[0049] In normal operation elementary decoder 1 (202) first receives the received data (110). Using the systematic data and the error correcting data aimed at elementary decoder 1 (202), the elementary decoder 1 (202) computes data

representing information on the original data (106). Typically this data will take the form of probabilities. Typically this data will be expressed in log-likelihood numbers. This type of data is called soft data. A person skilled in the art knows how a turbo decoder operates using soft data.

[0050] Next, elementary decoder 2 (204) receives the received data (110) and the output of elementary decoder 1 (202). Using the systematic data, the error correcting data aimed at elementary decoder 2 (204) and the output of elementary decoder 1 (202), the elementary decoder 2 (204) computes data representing information on the original data (106).

[0051] Next, elementary decoder 1 (202) receives the received data (110) and the output of elementary decoder 2 (204). Using the systematic data, the error correcting data aimed at elementary decoder 1 (202) and the output of elementary decoder 2 (204), the elementary decoder 1 (202) computes data representing information on the original data (106).

[0052] Turbo decoder system (112) continues to work in an iterative fashion, each elementary decoder taking as input the output of the other elementary decoder. On each iteration each elementary decoder computes a new output. Hence for each elementary decoder a sequence is created, a first sequence corresponding to the sequence of data computed by elementary decoder 1 (202) and a second sequence corresponding to the sequence of data computed by elementary decoder 2 (204).

[0053] In normal operation, if the error level in the received data (110), as compared with the transmitted data (109), is lower than the number of errors the turbo decoder system can correct, then the first sequence of outputs of the elementary decoder 1 (202) will converge. Likewise the second sequence of outputs of the elementary decoder 2 (204) will converge. Moreover they will converge to the same value. In normal circumstances the soft data of either elementary decoder will then represent the original data closely. At this point the soft data can be transformed into hard data. Whereas the soft data is a complicated and probabilistic representation of information on the original data, the hard data is a bit-pattern representing a best guess of the original data (106) on the basis of soft data. The transformation to hard data and the extraction of the hard data to produce decoded data (114) is not shown in Fig. 2.

[0054] The stopping decision unit (206) comprises at least one criterion or rule to decide when the iteration needs to be terminated. The stopping decision unit (206) comprises a method to terminate the elementary decoding operations based on the progress that is made towards a correct decoding. Either the stopping decision unit (206) may decide that the current set of soft data is, when transformed to hard data, likely a good enough approximation of the original data (106). Alternatively the stopping decision unit (206) may decide that the current set of soft data is, when transformed to hard data, likely a bad approximation of the original data (106), and moreover the soft data is not expected to become better in this regard. In both cases further elementary decodings are not needed and can be terminated. The embodiments below describe in further detail the operation of the stopping decision unit (206).

[0055] The stopping decision unit (206) can comprise more than one criterion to estimate the progress towards a correct decoding. A rule may also explicitly estimate the lack of progress made towards a correct decoding, or estimate the divergence. Combining more than one criterion makes it more likely that situations in which the elementary decoding can be stopped are recognized early. In addition rules that estimate the progress towards a correct decoding can be combined. For example, the stopping decision unit (206) may terminate if a fixed number of iterations have passed. This rule on the number of iterations can be combined with any other rule. Typically 8 iterations will suffice for most applications.

[0056] The turbo decoder system (112) can be made using dedicated hardware, such as electronic circuits that are configured to perform the method, or the turbo decoder system (112) can be made from generic hardware controlled using software, or the turbo decoder system (112) may comprise a combination of dedicated hardware, generic hardware and software to implement the method of the invention.

[0057] Since the stopping decision unit (206) has access to elementary decoder 1 (202) and elementary decoder 2 (204) it can terminate the elementary decodings at any point in the process. In particular, there is no reason to restrict the termination to the end of an iteration. As a result, on average, the turbo decoder will finish earlier. As a result the decoded data (114) is available earlier. Since fewer computations are done also the power consumption can be reduced. Alternatively, using the same power consumption and latency a larger turbo code can be used, e.g., a turbo code with a larger block size or a turbo code which can correct more errors, leading to larger bandwidth or better resilience against errors during transmission.

[0058] An advantage of deciding on termination using soft data instead of hard data can be that convergence patterns are earlier visible in the soft data. As a result a slightly more aggressive rule may be used. Since soft data represents quantized probability metrics but hard data corresponds instead to a binary hard decision made upon their sign, it is possible to express the point on which to terminate with much less complexity with respect to soft data.

[0059] The stopping decision unit (206) can terminate elementary decoder 1 (202) and elementary decoder 2 (204) by physically stopping them from decoding. After the termination the elementary decoder 1 (202) and elementary decoder 2 (204) are free to start decoding of a new received data, or may be left idle to save power consumption. Alternatively the stopping decision unit (206) can terminate elementary decoder 1 (202) and elementary decoder 2 (204) by causing the decoded data (114) to be extracted from the turbo decoder system (112) by a suitable extraction mechanism (not

shown). After the extraction it may be convenient for the architecture of the turbo decoder system (112) to leave elementary decoder 1 (202) and elementary decoder 2 (204) computing. The latter approach will improve latency but may be less suitable for power consumption reduction.

[0060] Fig. 3 describes an alternative embodiment of turbo decoder system (112). It contains all the elements of the figure 2 and, in addition, a hard data unit 1 (300) and a hard data unit 2 (302). Hard data unit 1 (300) is arranged to receive the output of elementary decoder 1 (202). The output of elementary decoder 1 (202) represents soft data and hard data unit 1 (300) is capable of converting the soft data into hard data. Likewise hard data unit 2 (302) is arranged to receive the output of elementary decoder 2 (204). The output of elementary decoder 2 (204) represents soft data and hard data unit 2 (302) is capable of converting it into hard data.

[0061] In this embodiment the stopping decision unit (206) is arranged to receive hard data from hard data unit 1 (300) and hard data unit 2 (302) instead of soft data.

[0062] An advantage of deciding on termination using hard data instead of soft data can be that hard data corresponds to the final output of the decoder; as a result it is easier to take into account consideration of processing steps done after the turbo decoding. Also considerations on hard data correspond better to numbers of errors and are in this regard better predictable.

[0063] Note that in general the stopping decision unit (206) may receive hard data for some elementary decoders and soft data for others. This may be advantageous if hard data is already available for some other reason. This arrangement may also have architectural advantages.

[0064] Fig.4 describes an embodiment of a stopping decision unit (206). The stopping decision unit (206) is arranged to receive output from elementary decoder 1 and output from elementary decoder 2, these outputs may either represent hard data or soft data. The stopping decision unit (206) comprises a memory 1 (400) and a memory 2 (402). In order to keep track of which data is needed and which memory can be used it is most advantageously to arrange both the memory 1 (400) and the memory 2 (402) as a queue, such as a First In First Out buffer (FIFO buffer). The stopping decision unit (206) comprises stopping decision logic (404).

[0065] The size of the memory 1 (400) and memory 2 (402) depend on the embodiment of the method chosen. In general the memories should be able to contain a representation of the soft data output of an elementary decoder, or the memories should be able to contain a representation of the hard data corresponding to the output of an elementary decoder. In case the stopping decision unit is configured to take into account more of the sequence of outputs of an elementary decoder the memory 1 (400) and memory 2 (402) need to be correspondingly larger.

[0066] In operation the stopping decision unit receives an output from elementary decoder 1 (202) or elementary decoder 2 (204). A representation of the received output from elementary decoder 1 (202) is stored in the memory 1 (400). A representation of the received output from elementary decoder 2 (204) is stored in the memory 2 (402). If necessary, information that is no longer needed for the application of the method can be overwritten.

[0067] As each new output is received the stopping decision logic (404) determines if the elementary decoding operations need to be terminated. It applies at least one criterion; for example, it applies the method of the invention. If the decoding operations have not been terminated yet it may also apply a number of other criteria. For example it may decide to terminate based on the number of iterations that happened so far. It may also apply other criteria.

[0068] The order in which the stopping decision logic (404) applies its criteria can be varied, e.g., it could first start using as a criterion the number of iterations and then apply other criteria, such as a method according to the invention.

[0069] The stopping decision unit may refrain from deciding to terminate during some elementary decoding operations, for example, it may skip the first or first few elementary decoding operations, as the chances of terminating the elementary decoding operations is small during those elementary decoding operations as compared to later elementary decoding operations.

[0070] Stopping decision logic (404) implements an embodiment of the method of the invention using dedicated hardware, such as electronic circuits that are configured to perform the method or the stopping decision logic (404) can be made from generic hardware controlled using software, or the stopping decision logic (404) may comprise a combination of dedicated hardware, generic hardware and software to implement the method of the invention.

[0071] Fig. 5 is a flowchart illustrating an embodiment (500) of the method of the invention. The starting point of the method is when any of the elementary decoders performs a specific elementary decoding operation (501). This specific elementary decoding operation was performed by a specific elementary decoder. To this specific elementary decoder corresponds a specific sequence of elementary decoding operations. The specific sequence of elementary decoding operations includes the specific elementary decoding operation (501). It is then determined if the specific sequence of elementary decoding operations has substantially converged or substantially diverged (502). If the specific sequence of elementary decoding operations has substantially converged or substantially diverged (502) then the embodiment decides to terminate (504) the elementary decoding operations, if not the test is applied after the next elementary decoding operation.

[0072] There are various ways in which the method can determine if the specific sequence has substantially converged or substantially diverged. These are discussed below.

[0073] Fig. 6 is a flowchart illustrating additional steps (600) that can be performed in conjunction with the method of the invention. The starting point of the additional steps is when any of the elementary decoders performs a specific elementary decoding operation (501). This specific elementary decoding operation was performed by a specific elementary decoder. To this specific elementary decoder corresponds a specific sequence of elementary decoding operations. The specific sequence of elementary decoding operations includes the specific elementary decoding operation (501). It is then determined if the specific sequence of elementary decoding operations has substantially converged (602). If so, it is determined whether the sequence of elementary decoding operations corresponding to the elementary decoder that was active previously to the specific elementary decoder has also converged (604). If so it is verified if both sequences converged to the same data (606). If both sequences did not converge to the same data then report a failure (608). In case both sequences converged, irrespective whether they converged to the same data or not, the method then terminates the decoding operations (504).

[0074] The advantage of these additional steps is that situations are recognized in which a turbo decoder is 'stuck'. If two or more sequences of elementary decoding operations have converged it is not likely that the situation will change by further computations, probably none of the converged data is likely to correspond to the original data (106).

[0075] As a result a turbo decoder system (112) in which this method is used avoids computing in situations that are already hopeless. By terminating the elementary decoding operations early a communication system comprising the turbo decoder system (112) has, e.g., the option to request a resend of the data quicker, thereby increasing the latency of such a communication system. Also power consumption is reduced since fewer computations are performed.

[0076] This approach is especially advantageous if it is verified that all sequences of elementary decoding operations have converged. As in that situation it is especially unlikely that further computations are useless.

[0077] Fig. 7 is a flowchart illustrating further additional steps (700) that can be performed in conjunction with the method of the invention. The starting point of the further additional steps is when any of the elementary decoders performs a specific elementary decoding operation (501). This specific elementary decoding operation was performed by a specific elementary decoder. To this specific elementary decoder corresponds a specific sequence of elementary decoding operations. The specific sequence of elementary decoding operations includes the specific elementary decoding operation (501). It is then determined if the specific sequence of elementary decoding operations substantially agrees (702) with the sequence of elementary decoding operations corresponding to the elementary decoder that was active previously to the specific elementary decoder. If so the decoding operations are terminated (504).

[0078] This embodiment shows a convenient way of finding two sequences that converge. After each elementary decoding, convergence is estimated of the corresponding sequence. In a memory (not shown) it is stored for each sequence if it has, so far, converged or not. By consulting the memory it can be determined if two sequences have converged. In this way, the method can be implemented without needing much memory. Alternative arrangements are also possible. For example, an embodiment could keep track of convergence only for two particular sequences corresponding to two particular elementary decoders, such as to the last two elementary decoders.

[0079] If the decoding operations were not terminated, then other reasons for terminating can be tried (704), such as those specified in the embodiments above.

[0080] The advantage of this method is that even if none or not all of the sequences have converged it is probably advantageous to terminate if two sequences substantially agree. It would be unlikely that two elementary decoders would agree by chance, it is more likely that they agree because both have found a good approximation to the original data (106).

[0081] An advantage of this method is that it combines different criteria for terminating the elementary decoding operations. Since testing for termination is relatively cheap compared to an extra elementary decoding operation it is opportune to test more than one criteria to maximize the probability of early stopping.

[0082] In general in the order of the various steps in all of these embodiments can be varied. In particular, steps (702) and (704) can be interchanged.

[0083] Fig. 8 gives various equations representing various aspects of embodiments of the method according to the invention based on hard data.

[0084] In all equations of figure 8, the letter 'X' refers to the sequence of elementary decoding operations corresponding to the specific elementary decoding operations (501). The letter 'A' refers to the sequence of elementary decoding operations corresponding to elementary decoder 1 (202). The letter 'B' refers to the sequence of elementary decoding operations corresponding to elementary decoder 2 (204). The index 'n' refers to the most recent elementary decoding operation of the elementary decoder it corresponds to. For example, 'A_n' refers to the most recent elementary decoding of elementary decoder 1 (202). Similarly 'A_{n-1}' refers to the second most recent (i.e. the one immediately preceding the most recent) elementary decoding of elementary decoder 1 (202).

[0085] It is noted that if an equation refers to a result that is not yet available, for example, 'A_{n-1}' in the first iteration, the method is simply skipped, and no termination is done for that reason.

[0086] All equations of figure 8 are related to hard data. The equality sign '=' means substantially equal. "Substantially equal" must be interpreted with respect to the block size of the turbo code, to the number of errors it can correct, and to the operations that may follow the turbo decoding. In case the turbo decoder is followed by a second other decoder,

such as a BCH decoder (a Bose-Chaudhuri-Hocquenghem decoder) the number of errors the BCH decoder can correct can be taken as a measure of substantially equal.

[0087] Using the equations of (801), (802) the method will terminate if the specific sequence has substantially converged. In (801) only the previous data is taken into account to determine convergence. In (802) more data is taken into account.

[0088] The equations of (803), (804) or (805) can be used to determine substantial agreement. As more data agrees, the sequences are more likely to be in real agreement as opposed to a chance agreement.

[0089] In the equations (801) to (805) the advantage of taking more data into account is that it becomes less likely to make a wrong termination decision. This will help the robustness. The advantage of taking fewer data into account is that termination will be decided on earlier. Deciding between these equations allows one to achieve a different balance between latency and robustness.

[0090] The equations (806) to (810) combine the previous equations in various ways, that is, they combine substantial convergence of the specific sequence with substantial agreement of two sequences. Since two criteria are combined when deciding upon termination, such a termination will, on average, happen earlier. If either criterion occurs the method decides to terminate the elementary decoding operations.

[0091] Note that since the elementary decoders need not all be of the same type, it may be advantageous to employ a different stopping rule after each elementary decoding step. For example, should it be known that one elementary decoder converges fast, for example, because it can correct more errors, it may be decided that it has converged, for example, by taking rule (801) for that elementary decoder but (802) for others.

[0092] Substantial divergence can be determined in similar ways. For example, if instead of equality in (802) there is a difference, and this difference is larger than a predetermined value, divergence may be assumed. Or, as an example, if in equation (802) instead of two equalities there are two differences and the more recent difference is larger than less recent difference.

[0093] Substantial agreement or substantial convergence can also be decided upon by operation upon soft data. For a person skilled in the art it is clear how the above equations can be adapted to use soft data. In particular, a sequence of elementary decoding operation can be considered converged if the sum of all bit reliabilities converges. Two sequences of elementary decoding operations can be considered to be in substantial agreement if the sum of the absolute value of the differences between the two sequences of the probabilities concerning the original data (106), such as comprised in soft data converges to 0. Instead of the sum other mathematical function can be used, preferably convex functions, such as a sum of powers, such as squares.

[0094] It should be evident to one with ordinary skills in the art that many combinations of these equations are possible, within the scope of the present invention.

[0095] Equation (801) can be used to establish substantial convergence and can be applied as follows. After each specific elementary decoding of a specific decoder (such as (202), (204) or any other elementary decoder that may be present) it is determined if the hard data corresponding to the specific elementary decoding (i.e. the recent decoding) is equal to the hard data corresponding to the second most recent (i.e. the one immediately preceding the most recent) decoding of the specific decoder. When it is such determined the execution of elementary decodings is terminated.

[0096] An equivalent formulation of equation (801) is given in equation (901). 'Equal' and '=' in equation (801), (901) or any other of the equations of Fig. 8 and Fig. 9 and the descriptions thereof can be embodied as exactly equal. 'Equal' can alternatively also be embodied as substantially equal.

[0097] Equation (802) can be used to establish substantial convergence and can be applied as follows. After each specific elementary decoding of a specific decoder (such as (202), (204) or any other elementary decoder that may be present) it is determined if the hard data corresponding to the recent, second most recent and third most recent elementary decoding of the specific decoder are equal. When it is such determined the execution of elementary decodings is terminated.

[0098] Equation (803) can be used to establish substantial agreement and can be applied as follows. After each specific elementary decoding of a specific decoder (such as (202), (204) or any other elementary decoder that may be present) it is determined if the hard data corresponding to the recent, elementary decoding of the specific decoder is equal to the hard data corresponding to the most recent elementary decoding of another elementary decoder. When substantial agreement is established the execution of elementary decodings is terminated.

[0099] Equation (804) can be used to establish substantial agreement and can be applied as follows. After each specific elementary decoding of a specific decoder (such as (202), (204) or any other elementary decoder that may be present) it is determined if the hard data corresponding to the specific elementary decoding (i.e. the recent decoding) is equal to the hard data corresponding to the second most recent (i.e. the one immediately preceding the most recent) decoding of the specific decoder. It is also determined if a recent particular decoding of a particular decoder (such as (202), (204) or any other elementary decoder that may be present) is equal to the hard data corresponding to the second most recent (i.e. the one immediately preceding the most recent) decoding of the particular decoder. Moreover it is determined if the recent specific elementary decoding is equal to the recent particular elementary decoding. The specific

decoder is not equal to the particular decoder. When substantial agreement is established the execution of elementary decodings is terminated.

[0100] Equation (805) can be used to establish substantial agreement and can be applied as follows. After each specific elementary decoding of a specific decoder (such as (202), (204) or any other elementary decoder that may be present) it is determined if the hard data corresponding to the specific elementary decoding (i.e. the recent decoding) is equal to the hard data corresponding to the second most recent (i.e. the one immediately preceding the most recent) decoding and to the third most recent of the specific decoder. It is also determined if a recent particular decoding of a particular decoder (such as (202), (204) or any other elementary decoder that may be present) is equal to the hard data corresponding to the second most recent (i.e. the one immediately preceding the most recent) decoding and to the third most recent of the particular decoder. Moreover it is determined if the recent specific elementary decoding is equal to the recent particular elementary decoding. The specific decoder is not equal to the particular decoder. When substantial agreement is established the execution of elementary decodings is terminated.

[0101] Equation (806) can be applied as a stopping criterion. When substantial convergence is established for one elementary decoder in the way as described for equation (801) or substantial agreement is established in the way as described for equation (803), the execution of elementary decodings is terminated. An equivalent formulation of equation (806) is given in equation (902).

[0102] Equation (807) can be applied as a stopping criterion. When substantial convergence is established for two elementary decoders in the way as described for equation (801) or substantial agreement is established in the way as described for equation (803) the execution of elementary decodings is terminated.

[0103] Equation (808) can be applied as a stopping criterion. When substantial convergence is established for two elementary decoders in the way as described for equation (801) or substantial agreement is established in the way as described for equation (804) the execution of elementary decodings is terminated.

[0104] Equation (809) can be applied as a stopping criterion. When substantial convergence is established for two elementary decoders in the way as described for equation (802) or substantial agreement is established in the way as described for equation (804) the execution of elementary decodings is terminated.

[0105] Equation (810) can be applied as a stopping criterion. When substantial convergence is established for two elementary decoders in the way as described for equation (802) or substantial agreement is established in the way as described for equation (805) the execution of elementary decodings is terminated.

[0106] The present invention, as described in embodiments herein, may be implemented using a programmed processor executing programming instructions that are broadly described above in flow chart form that can be stored on any suitable electronic storage medium. However, those skilled in the art will appreciate that the processes described above can be implemented in any number of variations and in many suitable programming languages without departing from the present invention. For example, the order of certain operations carried out can often be varied, additional operations can be added or operations can be deleted without departing from the invention. Error trapping, enhancements and variations can be added without departing from the present invention. Such variations are contemplated and considered equivalent.

[0107] The present invention could be implemented using special purpose hardware and/or dedicated processors. Similarly, general purpose computers, microprocessor based computers, digital signal processors, microcontrollers, dedicated processors, custom circuits, ASICs and/or dedicated hard wired logic may be used to construct alternative equivalent embodiments of the present invention.

[0108] Those skilled in the art will appreciate that the program steps and associated data used to implement the embodiments described above can be implemented using disc storage as well as other forms of storage, such as, for example, Read Only Memory (ROM) devices, Random Access Memory (RAM) devices, optical storage elements, magnetic storage elements, magneto-optical storage elements, flash memory and/or other equivalent storage technologies without departing from the present invention. Such alternative storage devices should be considered equivalents.

[0109] While the invention has been described in conjunction with specific embodiments, it is evident that many alternatives, modifications, permutations and variations will become apparent to those of ordinary skill in the art in light of the foregoing description. Accordingly, it is intended that the present invention embrace all such alternatives, modifications and variations as fall within the scope of the appended claims

Claims

1. A method (500) for stopping execution of a turbo decoder wherein said turbo decoder comprises

- a plurality of elementary decoders (202, 204) including at least a first elementary decoder (202) and a second elementary decoder (204), wherein:

each respective one of said elementary decoders (202, 204) is configured for performing a respective sequence of decoding operations on received data, each decoding operation generating soft data, and each respective one of said elementary decoders is arranged to receive the soft data generated by at least one other of said plurality of elementary decoders, said second elementary decoder (204) being arranged to receive the soft data generated by said first elementary decoder (202);

- a plurality of hard data units (300, 302) associated to said plurality of elementary decoder (202, 204), for the purpose of transforming said soft data into hard data; said method for stopping further comprising:

- comparing, after transformation into hard data by said hard data units (300, 302), the output - (A_n , A_{n-1}), (resp. B_n , B_{n-1}) - generated by said plurality of elementary decoders at the completion of a most recent and a second most recent iteration, which is the one immediately preceding the most recent, and upon detection of an equality of the most recent and the second most recent iteration terminating the execution of decoding operations;

characterized in that said method for stopping further comprises:

- determining (604) at the completion of the most recent elementary decoding operation of said first decoder (202), whether the hard data A_n produced by said first decoder (202) is equal to the hard data B_{n-1} produced by the second most recent elementary decoding of the other elementary decoder, and upon detection of an equality, terminating the execution of decoding operations.

2. The method as in claim 1, wherein said turbo decoder is followed by a second other decoder, such as a Bose-Chaudhuri-Hocquenghem BCH, decoder.

3. The method (600) for stopping execution of a turbo decoder as in any one of the preceding claims, comprising additional steps; wherein said additional steps comprise:

- determining if a first sequence of decoding operations performed on the received data by one of said elementary decoders has converged to a first data (602), and if so,
- determining if a second sequence of decoding operations performed on the received data by an elementary decoder in its previous iteration has converged to a second data (604), and if so,
- verifying that the first data and the second data are not equal (606), and if so,
- reporting that the turbo decoder has failed (608).

4. The method (700) for stopping execution of a turbo decoder as claimed in any one of the preceding claims, comprising further additional steps; wherein said further additional steps comprise:

- determining if a first sequence of decoding operations performed on the received data by one of said elementary decoders agrees with a second sequence of decoding operations (702) performed on the received data by an elementary decoder in its previous iteration and if so,
- terminating the execution of decoding operations (504)

5. The method (700) as defined in claim 4, wherein said agreement between two sequences is determined if the sum of the absolute value of the differences between the two sequences of the probabilities concerning the original data (106), such as comprised in the soft data converges to 0.

6. The method (700) as defined in claim 5 wherein said agreement between two sequences is determined if a convex function, such as a sum of powers, of the absolute value of the differences between the two sequences of the probabilities concerning the original data (106), such as comprised in the soft data converges to 0.

7. The method (700) as defined in any one of the preceding claims, wherein said plurality of elementary decoders includes at least said first elementary decoder (202) and said second elementary decoder (204) and wherein the termination of the execution of the decoding operations is determined in accordance with the combination on one of the following criteria:

$$(X_n = X_{n-1})$$

$$(X_n = X_{n-1} = X_{n-2})$$

$$(A_n = B_n)$$

$$(A_n = A_{n-1} = B_n = B_{n-1})$$

$$(A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$$

$$(X_n = X_{n-1}) \text{ OR } (A_n = B_n)$$

$$\begin{matrix} \text{2} & \blacksquare & ((A_n = A_{n-1}) & \blacksquare & \text{2} & ((A_n = A_{n-1}) & \text{AND } (B_n = B_{n-1})) & \text{OR } (A_n = B_n) \end{matrix}$$

$$\begin{matrix} \text{2} & \blacksquare & ((A_n = A_{n-1}) & \blacksquare & \text{2} & ((A_n = A_{n-1}) & \text{AND } (B_n = B_{n-1})) & \text{OR } (A_n = A_{n-1} = B_n = B_{n-1}) \end{matrix}$$

$$((A_n = A_{n-1} = A_{n-2}) \text{ AND } (B_n = B_{n-1} = B_{n-2})) \text{ OR } (A_n = A_{n-1} = B_n = B_{n-1})$$

$$((A_n = A_{n-1} = A_{n-2}) \text{ AND } (B_n = B_{n-1} = B_{n-2})) \text{ OR } (A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$$

where

'X' refers to the sequence of elementary decoding operations corresponding to the specific elementary decoding operations

'A' refers to the sequence of elementary decoding operations corresponding to said first elementary decoder (202)

'B' refers to the sequence of elementary decoding operations corresponding to said second elementary decoder 2 (204)

8. A turbo decoder system (112) comprising a

- a plurality of elementary decoders (202, 204), wherein:

each respective one of said elementary decoders is configured for performing a respective sequence of decoding operations, and

each respective one of said elementary decoders is arranged to receive a respective input from at least one other of said plurality of elementary decoders, and

a stopping decision unit (206);

characterized in that said stopping decision unit is at least configured to execute the method of any one of the claims 1 to 5.

9. Software for configuring a data processing system to be operable to stop the execution of a turbo decoder using the method of any one of the claims 1 to 7.

Patentansprüche

1. Verfahren (500) zum Stoppen der Ausführung eines Turbo-Decoders, wobei der Turbo-Decoder umfasst:

- eine Mehrzahl von Elementardecodern (202, 204), die mindestens einen ersten Elementardecoder (202) und einen zweiten Elementardecoder (204) umfassen, wobei:

jeder der Elementardecoder (202, 204) jeweils so konfiguriert ist, dass er eine jeweilige Folge von Decodieroperationen an empfangenen Daten ausführt, wobei jede Decodieroperation Soft-Daten generiert, und jeder der Elementardecoder jeweils so ausgelegt ist, dass er die Soft-Daten empfängt, die von mindestens einem anderen der Mehrzahl von Elementardecodern generiert wurden, wobei der zweite Elementardecoder (204) so ausgelegt ist, dass er die vom ersten Decoder (202) generierten Soft-Daten empfängt;

- eine Mehrzahl von Hard-Dateneinheiten (300, 302), die mit der Mehrzahl von Elementardecodern (202, 204) assoziiert ist, zum Umwandeln der Soft-Daten in Hard-Daten; wobei das Verfahren zum Stoppen ferner umfasst:

- Vergleichen nach der Umwandlung in Hard-Daten durch die Hard-Dateneinheiten (300, 302) der Ausgabe
- (A_n, A_{n-1}) , (bzw. B_n, B_{n-1}) -, die von der Mehrzahl von Elementardecodern generiert wird, bei Abschluss einer letzten und einer vorletzten Iteration, welche die eine ist, die der letzten unmittelbar vorangeht, und
- Beenden der Ausführung von Decodieroperationen bei Erkennung einer Gleichheit der letzten und der vorletzten Iteration;

dadurch gekennzeichnet, dass das Verfahren zum Stoppen ferner umfasst:

- Bestimmen (604) bei Abschluss der letzten elementaren Decodieroperation des ersten Decoders (202), ob die vom ersten Decoder (202) erzeugten Hard-Daten A_n gleich den durch die letzte elementare Decodieroperation des anderen Elementardecoders erzeugten Hard-Daten B_{n-1} sind, und
- Beenden der Ausführung von Decodieroperationen bei Erkennung einer Gleichheit.

2. Verfahren nach Anspruch 1, wobei auf den Turbo-Decoder ein zweiter, anderer Decoder, wie beispielsweise ein Bose-Chaudhuri-Hocquenghem, BCH,-Decoder, folgt.

3. Verfahren (600) zum Stoppen der Ausführung eines Turbo-Decoders nach einem der vorhergehenden Ansprüche, umfassend zusätzliche Schritte, wobei die zusätzlichen Schritte umfassen:

- Bestimmen, ob eine erste Folge von Decodieroperationen, die von einem der Elementardecoder an den empfangenen Daten ausgeführt werden, gegen erste Daten konvergiert hat (602), und gegebenenfalls
- Bestimmen, ob eine zweite Folge von Decodieroperationen, die von einem der Elementardecoder bei seiner vorherigen Iteration an den empfangenen Daten ausgeführt wurden, gegen zweite Daten konvergiert hat (604), und gegebenenfalls
- Verifizieren, dass die ersten Daten und die zweiten Daten nicht gleich sind (606), und gegebenenfalls
- Melden, dass der Turbo-Decoder versagt hat (608).

4. Verfahren (700) zum Stoppen der Ausführung eines Turbo-Decoders nach einem der vorhergehenden Ansprüche, umfassend weitere zusätzliche Schritte, wobei die weiteren zusätzlichen Schritte umfassen:

- Bestimmen, ob eine erste Folge von Decodieroperationen, die von einem der Elementardecoder an den empfangenen Daten ausgeführt werden, mit einer zweiten Folge von Decodieroperationen übereinstimmt (702), die von einem Elementardecoder bei seiner vorherigen Iteration an den empfangenen Daten ausgeführt wurden, und gegebenenfalls
- Beenden der Ausführung von Decodieroperationen (504).

5. Verfahren (700) nach Anspruch 4, wobei die Übereinstimmung zwischen zwei Folgen bestimmt wird, wenn die Summe des absoluten Wertes der Differenzen zwischen den beiden Folgen der Wahrscheinlichkeiten bezüglich der ursprünglichen Daten (106), wie in den Soft-Daten enthalten, gegen 0 konvergiert.

6. Verfahren (700) nach Anspruch 5, wobei die Übereinstimmung zwischen zwei Folgen bestimmt wird, wenn eine konvexe Funktion, wie beispielsweise eine Summe von Potenzen, des absoluten Wertes der Differenzen zwischen

den beiden Folgen der Wahrscheinlichkeiten bezüglich der ursprünglichen Daten (106), wie in den Soft-Daten enthalten, gegen 0 konvergiert.

7. Verfahren (700) nach einem der vorhergehenden Ansprüche, wobei die Mehrzahl von Elementardecodern mindestens den ersten Elementardecoder (202) und den zweiten Elementardecoder (204) umfasst, und wobei die Beendigung der Ausführung der Decodieroperationen gemäß der Kombination bei einem der folgenden Kriterien bestimmt wird:

$$(X_n = X_{n-1})$$

$$(X_n = X_{n-1} = X_{n-2})$$

$$(A_n = B_n)$$

$$(A_n = A_{n-1} = B_n = B_{n-1})$$

$$(A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$$

$$(X_n = X_{n-1}) \text{ ODER } (A_n = B_n)$$

$$((A_n = A_{n-1}) \text{ UND } (B_n = B_{n-1})) \text{ ODER } (A_n = B_n)$$

$$((A_n = A_{n-1}) \text{ UND } (B_n = B_{n-1})) \text{ ODER } (A_n = A_{n-1} = B_n = B_{n-1})$$

$$((A_n = A_{n-1} = A_{n-2}) \text{ UND } (B_n = B_{n-1} = B_{n-2})) \text{ ODER } (A_n = A_{n-1} = B_n = B_{n-1})$$

$$((A_n = A_{n-1} = A_{n-2}) \text{ UND } (B_n = B_{n-1} = B_{n-2})) \text{ ODER } (A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$$

wobei

'X' sich auf die Folge von elementaren Decodieroperationen bezieht, die den spezifischen elementaren Decodieroperationen entsprechen,
'A' sich auf die Folge von elementaren Decodieroperationen bezieht, die dem ersten Elementardecoder (202) entsprechen,
'B' sich auf die Folge von elementaren Decodieroperationen bezieht, die dem zweiten Elementardecoder 2 (204) entsprechen.

8. Turbo-Decoder-System (112), umfassend:

- eine Mehrzahl von Elementardecodern (202, 204), wobei:

jeder der Elementardecoder jeweils so konfiguriert ist, dass er eine jeweilige Folge von Decodieroperationen ausführt, und
jeder der Elementardecoder jeweils so ausgelegt ist, dass er eine jeweilige Eingabe von mindestens einem

anderen der Mehrzahl von Elementardecodern empfängt, und

- eine Stopp-Entscheidungseinheit (206);

dadurch gekennzeichnet, dass die Stopp-Entscheidungseinheit wenigstens so konfiguriert ist, dass sie das Verfahren nach einem der Ansprüche 1 bis 5 durchführt.

9. Software zum derartigen Konfigurieren eines Datenverarbeitungssystems, dass es so betrieben werden kann, dass es die Ausführung eines Turbo-Decoders unter Verwendung des Verfahrens nach einem der Ansprüche 1 bis 7 stoppt.

Revendications

1. Procédé (500) d'arrêt de l'exécution d'un turbo-décodeur, dans lequel ledit turbo-décodeur comprend :

- une pluralité de décodeurs élémentaires (202, 204) comprenant au moins un premier décodeur élémentaire (202) et un deuxième décodeur élémentaire (204), dans lequel :

chacun respectif desdits décodeurs élémentaires (202, 204) est configuré pour effectuer une séquence respective d'opérations de décodage sur des données reçues, chaque opération de décodage générant des données molles, et

chacun respectif desdits décodeurs élémentaires est agencé pour recevoir les données molles générées par au moins un autre de ladite pluralité de décodeurs élémentaires, ledit deuxième décodeur élémentaire (204) étant agencé pour recevoir les données molles générées par ledit premier décodeur élémentaire (202) ; et

- une pluralité d'unités de données dures (300, 302) associées à ladite pluralité de décodeurs élémentaires (202, 204), dans le but de transformer lesdites données molles en données dures ; ledit procédé d'arrêt comprenant en outre :

- la comparaison, après la transformation en données dures par lesdites unités de données dures (300, 302), des sorties - (An, An-1), (resp. Bn, Bn-1) ... - générées par ladite pluralité de décodeurs élémentaires à l'achèvement de l'itération la plus récente et de la deuxième itération la plus récente, qui est celle précédant immédiatement la plus récente, et

- à la détection d'une égalité de l'itération la plus récente et de la deuxième itération la plus récente, la terminaison de l'exécution d'opérations de décodage ;

caractérisé en ce que ledit procédé d'arrêt comprend en outre :

- la détermination (604), à l'achèvement de l'opération de décodage élémentaire la plus récente dudit premier décodeur (202), si les données dures An produites par ledit premier décodeur (202) sont égales aux données dures Bn-1 produites par la deuxième opération de décodage élémentaire la plus récente de l'autre décodeur élémentaire, et

- à la détection d'une égalité, la terminaison de l'exécution des opérations de décodage.

2. Procédé selon la revendication 1, dans lequel ledit turbo-décodeur est suivi par un deuxième autre décodeur, comme un décodeur Bose-Chaudhuri-Hocquenghem, BCH.

3. Procédé (600) d'arrêt de l'exécution d'un turbo-décodeur selon l'une quelconque des revendications précédentes, comprenant des étapes supplémentaires ; dans lequel lesdites étapes supplémentaires comprennent :

- la détermination si une première séquence d'opérations de décodage effectuées sur les données reçues par l'un desdits décodeurs élémentaires a convergé vers une première donnée (602) et, si tel est le cas,

- la détermination si une deuxième séquence d'opérations de décodage effectuées sur les données reçues par un décodeur élémentaire à son itération précédente a convergé vers une deuxième donnée (604) et, si tel est le cas,

- la vérification que la première donnée et la deuxième donnée ne sont pas égales (606) et, si tel est le cas,

- le signallement de l'échec (608) du turbo-décodeur.

4. Procédé (700) d'arrêt de l'exécution d'un turbo-décodeur selon l'une quelconque des revendications précédentes, comprenant d'autres étapes supplémentaires ; dans lequel lesdites autres étapes supplémentaires comprennent :

- la détermination si une première séquence d'opérations de décodage effectuées sur les données reçues par l'un desdits décodeurs élémentaires s'accorde avec une deuxième séquence d'opérations de décodage (702) effectuées sur les données reçues par un décodeur élémentaire à son itération précédente et, si tel est le cas, - la terminaison de l'exécution des opérations de décodage (504).

5. Procédé (700) selon la revendication 4, dans lequel ledit accord entre deux séquences est déterminé si la somme de la valeur absolue des différences entre les deux séquences des probabilités concernant les données d'origine (106), telles qu'elles sont comprises dans les données molles, converge vers 0.

6. Procédé (700) selon la revendication 5, dans lequel ledit accord entre deux séquences est déterminé si une fonction convexe, comme une somme de puissances, de la valeur absolue des différences entre les deux séquences des probabilités concernant les données d'origine (106), telles qu'elles sont comprises dans les données molles, converge vers 0.

7. Procédé (700) selon l'une quelconque des revendications précédentes, dans lequel ladite pluralité de décodeurs élémentaires comprend au moins ledit premier décodeur élémentaire (202) et ledit deuxième décodeur élémentaire (204) et dans lequel la terminaison de l'exécution des opérations de décodage est déterminée en fonction de la combinaison à l'un des critères suivants :

$$(X_n = X_{n-1})$$

$$(X_n = X_{n-1} = X_{n-2})$$

$$(A_n = B_n)$$

$$(A_n = A_{n-1} = B_n = B_{n-1})$$

$$(A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$$

$$(X_n = X_{n-1}) \text{ OU } (A_n = B_n)$$

$$((A_n = A_{n-1}) \text{ ET } (B_n = B_{n-1})) \text{ OU } (A_n = B_n)$$

$$((A_n = A_{n-1}) \text{ ET } (B_n = B_{n-1})) \text{ OU } (A_n = A_{n-1} = B_n = B_{n-1})$$

$$((A_n = A_{n-1} = A_{n-2}) \text{ ET } (B_n = B_{n-1} = B_{n-2})) \text{ OU } (A_n = A_{n-1} = B_n = B_{n-1})$$

($A_n = A_{n-1} = A_{n-2}$) ET ($B_n = B_{n-1} = B_{n-2}$) OU ($A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2}$)

5 où

« X » fait référence à la séquence d'opérations de décodage élémentaire correspondant aux opérations de décodage élémentaire spécifiques ;

10 « A » fait référence à la séquence d'opérations de décodage élémentaire correspondant au dit premier décodeur élémentaire (202) ;

« B » fait référence à la séquence d'opérations de décodage élémentaire correspondant au dit deuxième décodeur élémentaire (204).

8. Système de turbo-décodeur (112) comprenant :

15 - une pluralité de décodeurs élémentaires (202, 204), dans lequel :

chacun respectif desdits décodeurs élémentaires est configuré pour effectuer une séquence respective d'opérations de décodage, et

20 chacun respectif desdits décodeurs élémentaires est agencé pour recevoir une entrée respective d'au moins un autre de ladite pluralité de décodeurs élémentaires, et

- une unité de décision d'arrêt (206) ;

25 **caractérisé en ce que** ladite unité de décision d'arrêt est au moins configurée pour exécuter le procédé selon l'une quelconque des revendications 1 à 5.

9. Logiciel pour configurer un système de traitement de données afin qu'il soit utilisable pour arrêter l'exécution d'un turbo-décodeur en utilisant le procédé selon l'une quelconque des revendications 1 à 7.

30

35

40

45

50

55

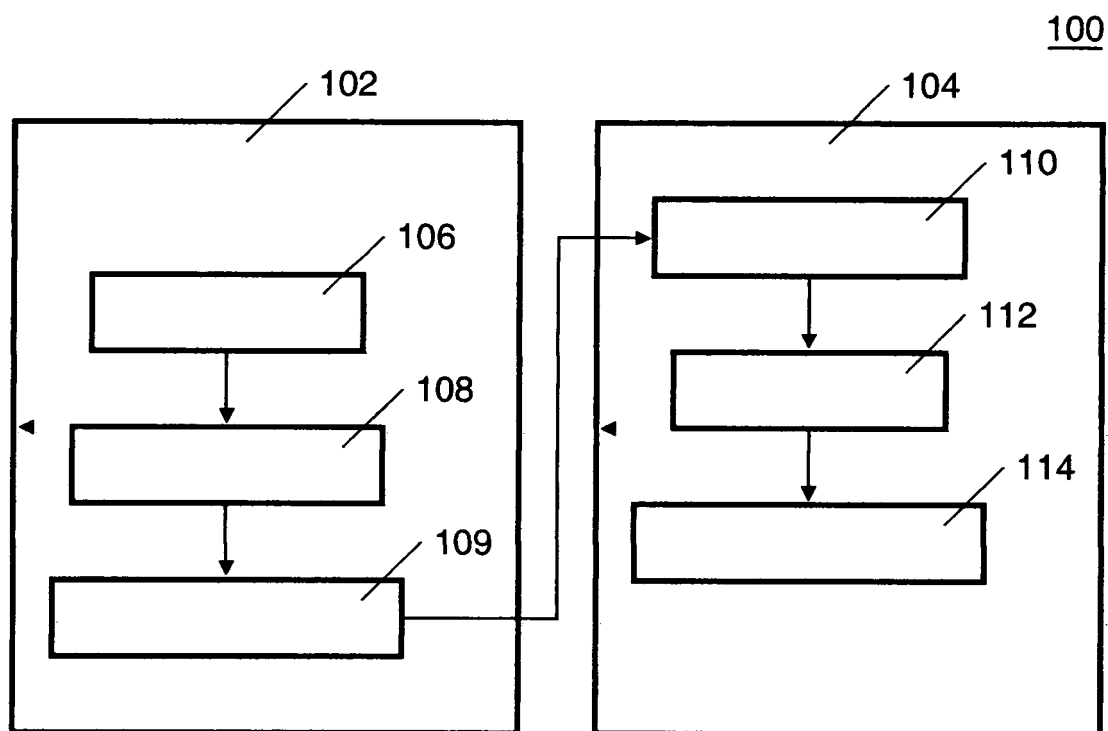


FIG. 1

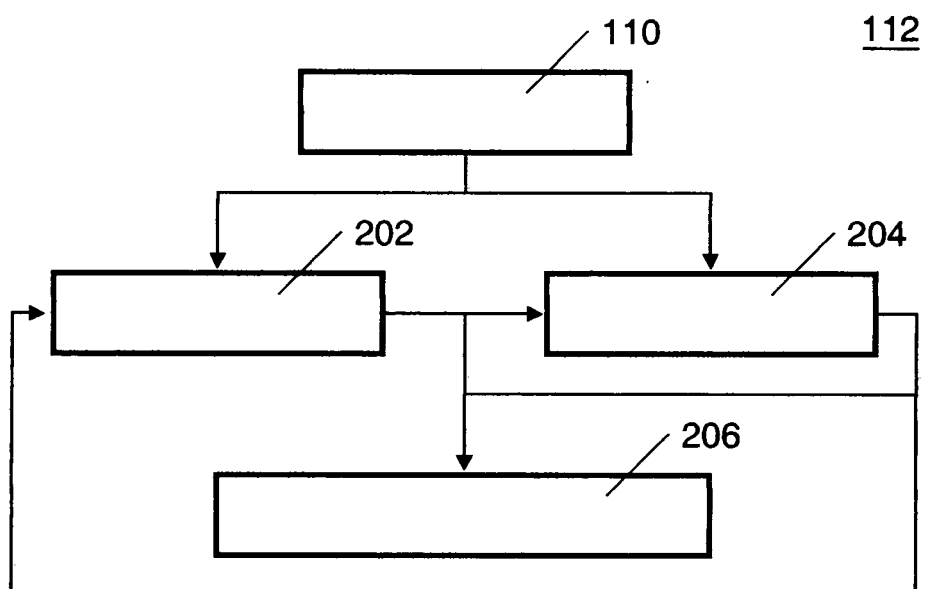


FIG. 2

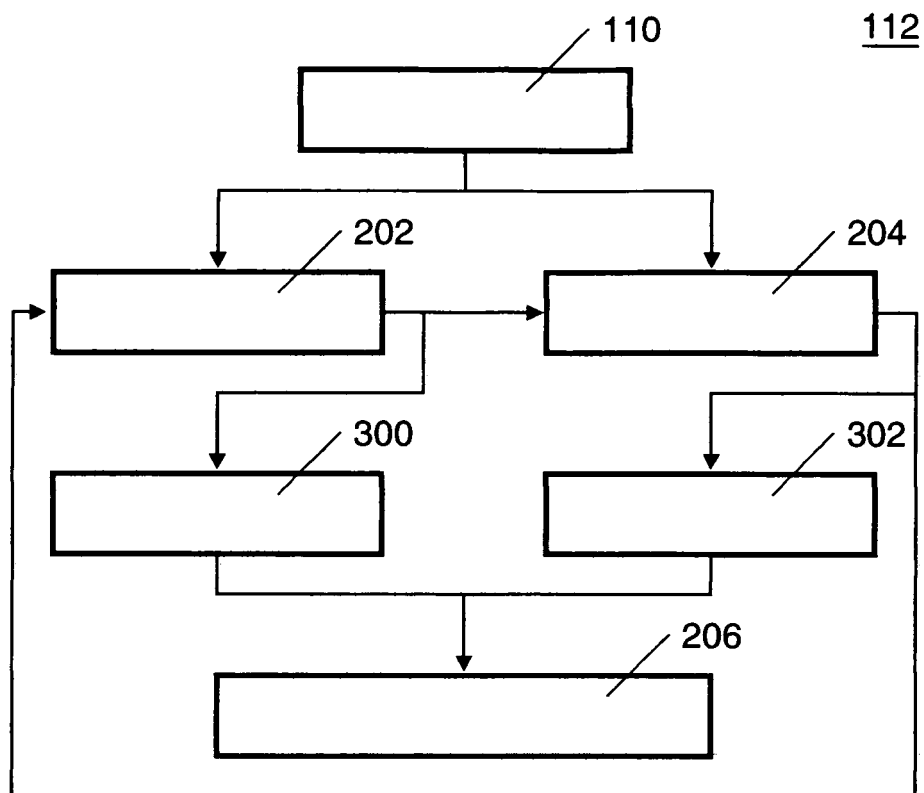


FIG. 3

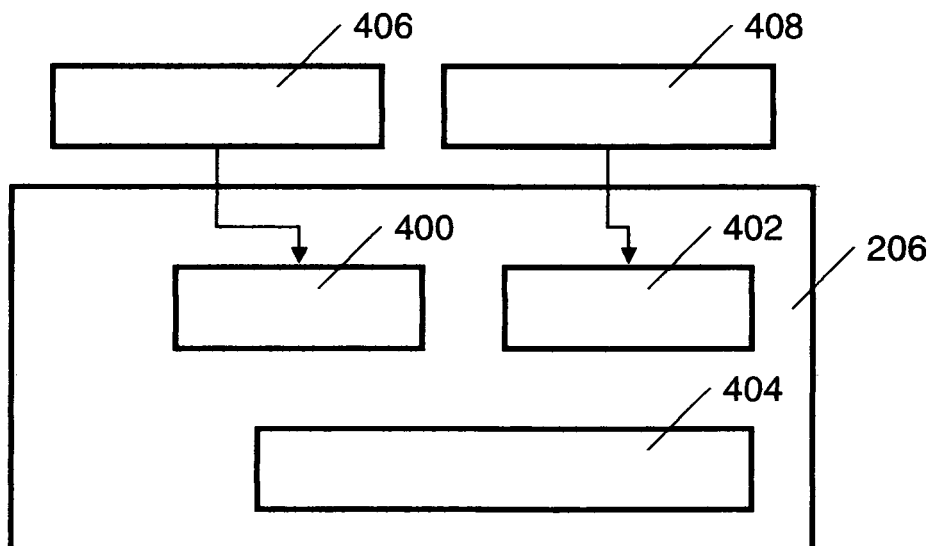


FIG. 4

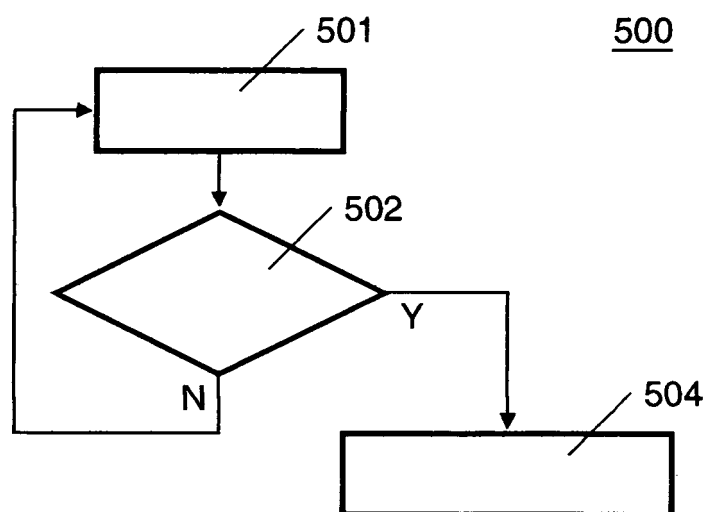


FIG. 5

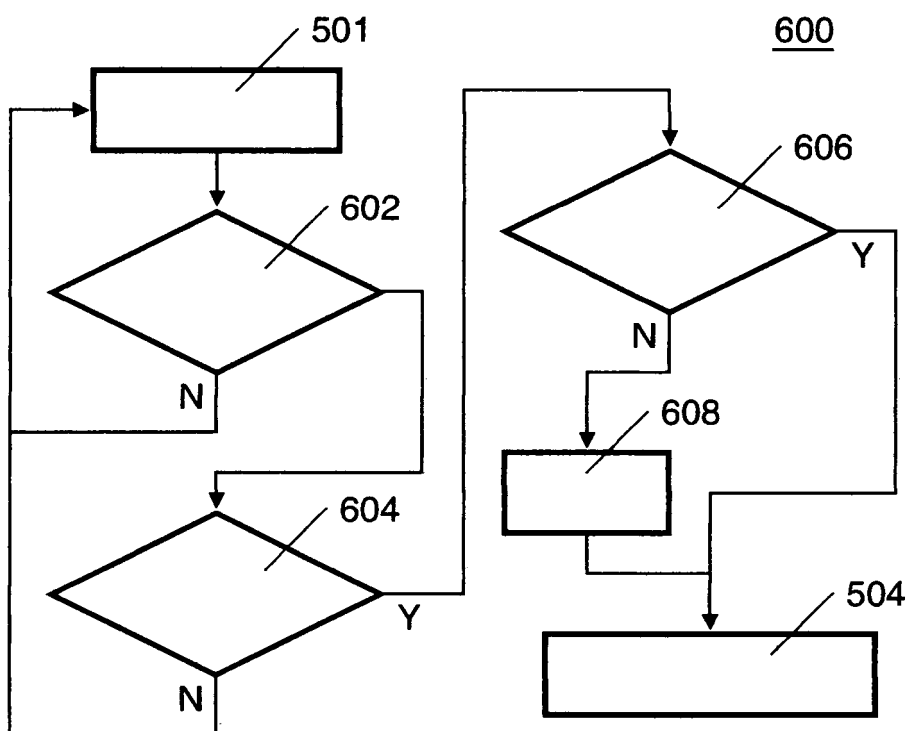


FIG. 6

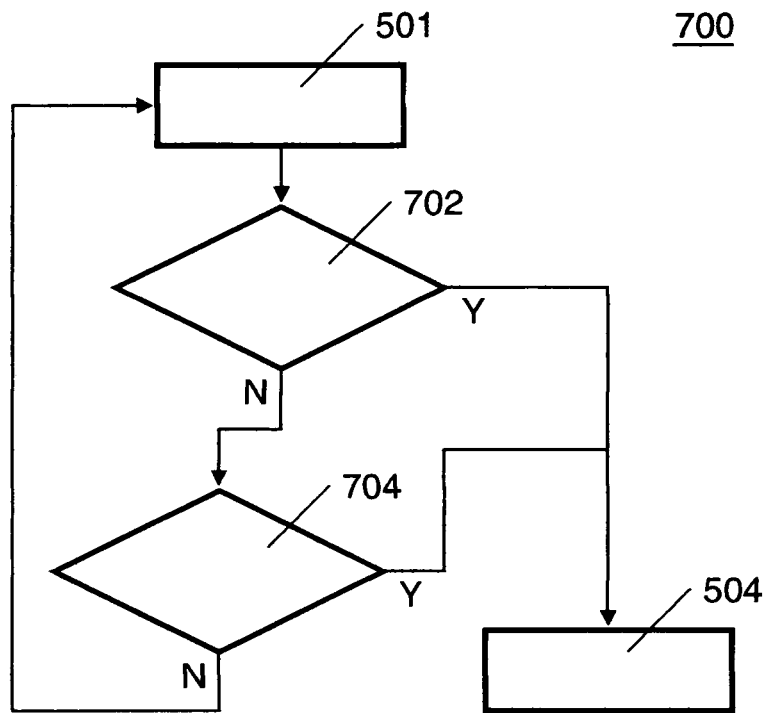


FIG. 7

801. $(X_n = X_{n-1})$
 802. $(X_n = X_{n-1} = X_{n-2})$
 803. $(A_n = B_n)$
 804. $(A_n = A_{n-1} = B_n = B_{n-1})$
 805. $(A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$
 806. $(X_n = X_{n-1}) \text{ OR } (A_n = B_n)$
 807. $((A_n = A_{n-1}) \text{ AND } (B_n = B_{n-1})) \text{ OR } (A_n = B_n)$
 808. $((A_n = A_{n-1}) \text{ AND } (B_n = B_{n-1})) \text{ OR } (A_n = A_{n-1} = B_n = B_{n-1})$
 809. $((A_n = A_{n-1} = A_{n-2}) \text{ AND } (B_n = B_{n-1} = B_{n-2})) \text{ OR } (A_n = A_{n-1} = B_n = B_{n-1})$
 810. $((A_n = A_{n-1} = A_{n-2}) \text{ AND } (B_n = B_{n-1} = B_{n-2})) \text{ OR } (A_n = A_{n-1} = A_{n-2} = B_n = B_{n-1} = B_{n-2})$

FIG. 8

901. $(A_n = A_{n-1}) \text{ or } (B_n = B_{n-1})$
 902. I: $(A_n = A_{n-1}) \text{ OR } (A_n = B_n)$
 II: $(B_n = B_{n-1}) \text{ OR } (A_n = B_n)$

FIG. 9

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 7093180 B2 [0010]
- EP 1217776 A [0012]

Non-patent literature cited in the description

- **S. A. BARBULESCU ; S. S. PIETROBON.** Turbo codes: A tutorial on a new class of powerful error correcting coding schemes, Part 1: Code structures and interleaver design. *J. Elec. and Electron. Eng., Australia*, September 1999, vol. 19, 129-142 [0008]
- **S. A. BARBULESCU ; S. S. PIETROBON.** Turbo codes: A tutorial on a new class of powerful error correcting coding schemes, Part 2: Decoder design and performance. *J. Elec. and Electron. Eng., Australia*, September 1999, vol. 19, 143-152 [0008]
- **A. MATAACHE ; S. DOLINAR ; F. POLLARA.** Stopping Rules for Turbo Decoders. *TMO progress report 42-142*, 15 August 2000 [0011]